

CLEAN VERSION OF PARTIAL SPECIFICATION AND
ALL PENDING CLAIMS

In the Specification:

Please replace the title located on page 1, line 1, and page 66, line 1, with the following title: "METHOD FOR BUILDING A REAL-TIME CONTROL SYSTEM WITH MODE AND LOGICAL RATE".

Please replace the abstract located on page 66, lines 4-24, with the following abstract:

A development tool combines advantages of a simulation tool with an object-oriented modeling tool, including a real-time mathematical matrix library and an object model. The tool is applicable to any intelligent control system. A composite object group (COG) contains both sampled-data and event-driven capabilities. Mapping the computing resources of a computing device throughout the hierarchy of a control system is provided for modes, executable programs and threads of a processor. Within a given processor, each component of a control system may be assigned to individual execution threads within that processor simply by naming a thread.

In the Claims:

1. A composite object group (COG) data structure embodied in a computer-readable medium for building a control system that has a clock cycle, said COG data structure being implemented using object-oriented programming and comprising:

an interface for passing information to and from said COG data structure, said interface passing at least one control signal for controlling an aspect of said control system;

a data flow object arranged to accept input data and to produce output data on said clock cycle, said data flow object connected to said interface and providing sampled-data processing for said control system; and

a state machine object that includes a plurality of states and a plurality of transitions between said states that are each triggered by an event, said state machine object providing event-driven processing for said control system and being connected to said data flow object, whereby said COG data structure provides both sampled-data and event-driven processing for said control system.

A

2. A COG data structure as recited in claim 1 wherein said control system is an electromechanical control system.
3. A COG data structure as recited in claim 1 wherein said control system is a real-time control system.
4. A COG data structure as recited in claim 1 further comprising:

a state transition object included within said state machine object, said state transition object being associated with one of said transitions and providing an action to be performed upon said one of said transitions, whereby said state machine object provides said action for said COG.
5. A COG data structure as recited in claim 1 further comprising:

an execution method included in said data flow object, said execution method receiving said input data from sensors of said control system and calculating said output data.
6. A COG data structure as recited in claim 1 further comprising:

connections between said interface, said data flow object and said state machine object that are represented graphically to a user on a computer screen.
7. A COG data structure as recited in claim 1 further comprising:

a plurality of other COG data structures in a hierarchy, said hierarchy modeling said control system.
8. A computer-implemented method of defining a composite object group for use in building a control system, said method comprising:

A

providing an interface for passing information to and from said composite object group, said interface passing at least one control signal for controlling an aspect of said control system;

providing a data flow object that is arranged to accept input data and to produce output data on a clock cycle of said control system, said data flow object providing sampled-data processing for said electromechanical control system;

providing a state machine object that includes a plurality of states, said state machine object providing event-driven processing for said control system;

providing at least one transition between said states that is triggered by an event;

linking said state machine object to said data flow object; and

linking said state machine object or said data flow object to said interface, whereby said composite object group provides both sampled-data and event-driven processing for said control system.

9. A method as recited in claim 8 wherein said control system is an electromechanical control system.

10. A method as recited in claim 8 wherein said control system is a real-time control system.

11. A method as recited in claim 8 further comprising:

providing a state transition object, said state transition object being associated with said transition and providing an action to be performed upon transition, whereby said state transition object provides said action for said composite object group.

12. A method as recited in claim 8 further comprising:

defining an execution method of said data flow object, said execution method receiving said input data from sensors of said control system and calculating said output data.

13. A method as recited in claim 8 further comprising:

linking said state machine object to said data flow object using a graphical user interface;
and

linking said state machine object or said data flow object to said interface using a graphical user interface, whereby said composite object group is represented graphically to a user on a computer screen.

14. A method as recited in claim 8 further comprising:

defining a plurality of other composite object group in a hierarchy, said hierarchy modeling said control system.

15. A control system for use on a real-time computer, said control system comprising:

a system diagram graphically representing components and interconnections between components, said system diagram including

a data flow component arranged to accept input data and to produce output data on a clock cycle, said data flow component providing sampled-data processing for said control system, and

a state machine component that includes a plurality of states and a plurality of transitions between said states that are each triggered by an event, said state machine component providing event-driven processing for said control system; and

a database repository storing computer code associated with said system diagram and said components, said database repository including

a system diagram file representing said graphical system diagram, and

a compiled code file for each component including compiled computer code for executing the functionality of said each component, whereby said system diagram file and said compiled code files of said control system may be loaded onto said real-time computer for execution.

16. A control system as recited in claim 15 wherein said control system is an electromechanical control system.

17. A control system as recited in claim 15 wherein said control system is a real-time control system.

18. A control system as recited in claim 15 further comprising:

a composite object group included in said system diagram, said data flow component and said state machine component included in said composite object group, whereby said composite object group provides both sampled-data and event-driven processing for said control system; and

a diagram file for said composite object group included in said database repository representing the contents of said composite object group.

19. A method for building a control system for use on a real-time computer, said method comprising:

creating a data flow diagram arranged to accept input data and to produce output data on a clock cycle, said data flow diagram providing sampled-data processing for said control system;

creating a state machine component that includes a plurality of states and a plurality of transitions between said states that are each triggered by an event, said state machine component providing event-driven processing for said control system;

creating a system diagram that graphically represents said data flow diagram and said state machine component;

compiling said data flow diagram and said state machine component; and

loading said system diagram and compiled code for said data flow diagram and said state machine component onto said real-time computer, whereby said control system may be executed on said real-time computer.

20. A method as recited in claim 19 further comprising:

linking said compiled code using said system diagram on said real-time computer.

21. A control system as recited in claim 19 wherein said control system is an electromechanical control system.

22. A method for executing a control system on a real-time computer, said method comprising:

identifying a plurality of data flow components of said control system, each data flow component arranged to accept input data and to produce output data on a clock cycle, said data flow components providing sampled-data processing for said control system;

executing a method of each data flow component to produce said output data;

receiving an event for said control system;

matching said event with a transition of a state machine component of said control system, said state machine component including a plurality of states and a plurality of transitions between said states that are each triggered by an event, said state machine component providing event-driven processing for said control system; and

executing said matched transition of said state machine component.

23. A method of defining a mode of a control system, said method comprising:

receiving a plurality of components that represent said control system;

choosing a subset of said components to correspond to said mode;

identifying an intuitive name for said mode and its corresponding subset of said components; and

for each chosen component in said subset, indicating in a file corresponding to said component that said component is active when said control system is in said mode, whereby when said control system is executing in said mode, only said subset of said components that correspond to said mode will be active.

24. A method as recited in claim 23 wherein said control system is an electromechanical control system.

25. A method as recited in claim 23 wherein said control system is a real-time control system.

26. (Cancelled).

27. A method as recited in claim 23 wherein said received components represent a low level in a multi-level hierarchy of said control system, said method further comprising:

presenting a high level of said hierarchy, said high level including said received components and a high level component;

choosing a second mode for said high level component; and

choosing a third mode for said control system that includes said mode and said second mode, whereby when said control system is executing in said third mode, only those components that correspond to said mode and said second mode will be active.

28. A method as recited in claim 23 wherein said components are computer objects and said control system is implemented using object-oriented technology.

29. A method as recited in claim 23 wherein said components are loaded into a real-time computer, said method further comprising:

executing said control system in said mode on said real-time computer; and

activating only those components which correspond to said mode.

30. A method as recited in claim 29 further comprising:
- switching from said mode to a different mode while said control system is executing;
- deactivating those components that correspond to said mode; and
- activating those components that correspond to said different mode.
31. A method of mapping a thread of a processor to components within a control system, said method comprising:
- receiving a plurality of components in a multi-level hierarchy that represent said control system;
- defining a logical rate of execution at each level in said multi-level hierarchy;
- for each component in a level of said multi-level hierarchy, assigning said component to one of said logical rates; and
- mapping said thread of said processor through said multi-level hierarchy such that at least one of said components is assigned to said thread, whereby an execution rate of said thread is assigned to said least one component.
32. A method as recited in claim 31 wherein said control system is an electromechanical control system.
33. A method as recited in claim 31 wherein said control system is a real-time control system.
34. (Cancelled).
35. A method as recited in claim 31 wherein said components are computer objects and said control system is implemented using object-oriented technology.

36. A method as recited in claim 31 wherein said components are loaded into a real-time computer, said method further comprising:

executing said control system on said real-time computer; and

executing said at least one component at said rate of execution which corresponds to said thread.

37. A method of mapping threads available on a real-time computer to components within a control system, said method comprising:

receiving a plurality of components in a multi-level hierarchy that represent said control system;

defining at least one logical rate of execution at each level in said multi-level hierarchy;

for each component in a level of said multi-level hierarchy, assigning said component to one of said logical rates; and

mapping said threads of said real-time computer through said multi-level hierarchy using said logical rates such that each of said components is assigned to a single one of said threads, whereby an execution rate of each thread is assigned to each one of said components.

38. A method of defining an executable image for a control system; said method comprising:

receiving a plurality of components that represent said control system;

choosing a subset of said components to correspond to said executable image;

identifying a name for said executable image and its corresponding subset of said components; and

for each chosen component in said subset, indicating in a file corresponding to said component that said component is part of said executable image for said control system, whereby when said executable image is produced for said control system only said subset of said components that correspond to said executable image will be included.

39. A method as recited in claim 38 wherein said control system is an electromechanical control system.

40. A method as recited in claim 38 wherein said control system is a real-time control system.

41. (Cancelled).

42. A method as recited in claim 38 wherein said received components represent a low level in a multi-level hierarchy of said control system, said method further comprising:

presenting a high level of said hierarchy, said high level including said received components and a high level component;

choosing a second executable image for said high level component; and

choosing a third executable image for said control system that includes said executable image and said second executable image, whereby when said third executable image is produced for said control system only those components that correspond to said executable image and said second executable image will be included.

43. A method as recited in claim 38 wherein said components are computer objects and said control system is implemented using object-oriented technology.

44. A method as recited in claim 38 further comprising:

loading components corresponding to said executable image into a real-time computer;
and

executing said executable image on said real-time computer.

45. A method of mapping an executable image destined for a particular computer to components within a control system, said method comprising:

receiving a plurality of components in a multi-level hierarchy that represent said control system;

defining at least one logical executable name at each level in said multi-level hierarchy;

for each component in a level of said multi-level hierarchy, assigning said component to one of said logical executable names; and

mapping said executable image destined for said particular computer through said multi-level hierarchy using said logical executable names such that a subset of said components are assigned to said executable image, whereby said executable image includes said subset of components.

46. A reusable interface component data structure for use in a control system having a plurality of components, said reusable interface component data structure comprising:

an interface component name, said name indicating an association with one of said components of said control system;

a group of first method names indicating methods defined upon said component that said component provides;

a group of second method names indicating methods defined within said control system that said component uses;

a group of output pin names indicating attributes defined upon said component for which said component provides values;

a group of input pin names indicating attributes defined within said control system whose values said component uses; whereby said interface component may be used as an interface to said component of said control system and reused with other of said components.

47. A reusable interface component data structure as recited in claim 46 wherein said control system is an electromechanical control system.

48. A reusable interface component data structure as recited in claim 46 wherein said control system is a real-time control system.

49. A reusable interface component data structure as recited in claim 46 further comprising:
a graphical representation, whereby said interface component appears as an icon to a user on a computer screen.

50. A reusable interface component data structure as recited in claim 46 wherein said interface component is a computer object and is implemented using object-oriented technology.